said texture cache memory in accordance with a replacement policy, and a direct memory access engine that retrieves texel data from memory.

52. (Amended) A texture mapping method using a graphics accelerator, said graphics accelerator including a texture cache memory, a cache controller that controls the texture cache memory in accordance with a cache replacement policy, and a direct memory access engine, comprising:

(a) retrieving texels from memory via the direct memory access engine;

(b) storing said retrieved texels in the texture cache memory in accordance with a replacement policy that is determined by the cache controller; and

(c) rendering a polygon using texels that are stored in the texture cache memory.

56. The texture mapping method of claim 55, wherein said storing comprises reviewing a state of flags within at least one set of flags that are associated with cache lines of said texture cache memory.

62. (Amended) A computer system, comprising:

a memory; and

a memory control that stores two-dimensional data in said memory, wherein said data is stored in said memory using an interleaved address that is formed by interleaving individual bit values of a coordinate in a first dimension with individual bit values of a coordinate in a second dimension, wherein at least part of said interleaved address has a

bit pattern that includes multiple bit values of said first dimension coordinate at odd-numbered bit positions and multiple bit values of said second dimension coordinate at even-numbered bit positions.

66. (Amended) A texture caching method, comprising:

(a) identifying a set of two-dimensional data that is to be transferred into memory; and

(b) storing said set of two-dimensional data in memory using an interleaved address that is formed by interleaving individual bit values of a coordinate in a first dimension with individual bit values of a coordinate in a second dimension, wherein at least part of said interleaved address has a bit pattern that includes multiple bit values of said first dimension coordinate at odd-numbered bit positions and multiple bit values of said second dimension coordinate at even-numbered bit positions.

69. The method of claim 68, wherein said storing comprises storing texels in linear cache lines of a texture cache in said graphics accelerator.

Please add new claims 70-89 as follows:

70. A graphics processing apparatus, comprising:

a graphics accelerator, said graphics accelerator, including

a texture cache memory;

a direct memory access engine coupled to a bus, wherein texel data is retrieved, by said direct memory access engine, from a memory over said bus and provided to said texture cache memory for storage; and

a texture engine coupled to said texture cache memory; said texture engine receiving texels from said texture cache memory to produce texture values for pixels,

wherein said texture cache memory is controlled by a texture cache controller resident on said graphics accelerator, said texture cache controller controlling said texture cache memory in accordance with a texture cache replacement policy.

71. A graphics accelerator for use in a computer system having a central processing unit, comprising:

a texture cache memory;

a direct memory access circuit that retrieves texel data from memory;

a texture value generating circuit coupled to said texture cache memory; said texture value generating circuit producing texture values for pixels based on texel data stored in said texture cache memory; and

a cache controller that controls said texture cache memory in accordance with a texture cache replacement policy.

72. The graphics accelerator of claim 71, wherein said texture cache is fully associative.

73. The graphics accelerator of claim 71, wherein said texture cache replacement policy is a least recently loaded policy.

74. The graphics accelerator of claim 71, wherein said texture cache replacement policy operates such that cache lines containing texels that are being used to compute texture values to describe a polygon cannot be overwritten until said polygon is complete.

75. The graphics accelerator of claim 74, wherein said texture cache replacement policy is implemented using at least one set of flags that are associated with cache lines of said texture cache.

76. The graphics accelerator of claim 71, wherein said direct memory access engine implements a virtual-physical address translation.

77. The graphics accelerator of claim 71, wherein said texture cache system is capable of operating in a prefetch mode such that during the rendering of a first polygon, a set of texels including at least those texels needed for completely rendering a second polygon are prefetched and stored in said texture cache memory.

78. The graphics accelerator of claim 77, wherein said set of texels are prefetched if it is determined that said set of texels can fit into space available in said texture cache memory.

79. The graphics accelerator of claim 78, wherein said set of texels are prefetched if it is determined that said set of texels can fit into one half of said texture cache memory.

80. The graphics accelerator of claim 77, wherein said texture cache operates in an on demand mode for said second polygon if said set of texels cannot be prefetched.

81. The graphics accelerator of claim 71, wherein said graphics accelerator is a graphics accelerator board.

82. A texture caching method, comprising:

(a) identifying a set of two-dimensional data that is to be transferred into memory; and

(b) storing said set of two-dimensional data in memory using an address, wherein at least part of said address is of the form U_NU_{N-1} . . . U_2U_1, wherein even-numbered bits of said address represent at least part of a first dimension coordinate and odd-numbered bits of said address represent at least part of a second dimension coordinate.

83. The texture caching method of claim 82, wherein said storing comprises storing texels in said memory.

84. The texture caching method of claim 83, wherein said storing comprises storing texels in memory of a graphics accelerator.

85. The texture caching method of claim 84, wherein said storing comprises storing texels in linear cache lines of a texture cache in said graphics accelerator.

86. A memory system, comprising:

a memory; and

a memory control that stores two-dimensional data in said memory, wherein said data is stored in said memory using an address, wherein at least part of said address is of the form $U_N U_{N-1} \ldots U_2 U_1$, wherein even-numbered bits of said address represent at least part of a first dimension coordinate and odd-numbered bits of said address represent at least part of a second dimension coordinate.

87. The memory system of claim 86, wherein said memory and memory control are included as part of a graphics accelerator, said memory control being operative to control the storage of texels in a texture cache.

88. The memory system of claim 87, wherein said texels are stored in linear cache lines of said texture cache.

89. The memory system of claim 88, wherein said texture cache is fully associative.

Dated: 5/14/01

Respectfully submitted,
COOLEY GODWARD LLP

By: _____
Duane S. Kobayashi
Reg. No. 41,122

Cooley Godward LLP
Attn: Patent Group
One Freedom Square - Reston Town Center
11951 Freedom Drive
Reston, VA 20190-5601
Tel: (703) 456-8000
Fax: (703) 456-8100
DSK:krf